◆ 実 践 報 告

なぜ計算機で足し算をできるのか ―「数学と人間の活動」の話題として

Why a computer can perform addition? As a topic of "mathematics and human activity"

理学部　数理科学科
鈴木　登志雄

1　序

平成３０年告示高等学校学習指導要領 [文科省 2019] には，教科「数学」の科目「数学 A」に「数学と人間活動」,「数学 B」に「数学と社会生活」，そして「数学 C」に「数学的な表現の工夫」という項目がある．これらは従前学習指導要領における科目「数学活用」の内容を継承し発展させたものである．「数学と人間活動」,「数学と社会生活」，および「数学的な表現の工夫」の説明中には，それぞれ以下の一節がある．

> [文科省 2019, p.98] 数量や図形に関する概念などと人間の活動との関わりについて理解すること．
>
> [文科省 2019, p.100] 日常の事象や社会の事象などを数学化し，数理的に問題を解決する方法を知ること．
>
> [文科省 2019, p.102] 日常の事象や社会の事象などを，離散グラフや行列を用いて工夫して表現することの意義を理解すること．

さて，筆者は 2006 年度，2012 年度，および 2018 年度に首都大学東京（現　東京都立大学）の数理科学教室が主催する「オープンクラス 高校生のための数学 ― 夏の学校」の講師を務めた．とくに 2018 年度に『なぜコンピューターで自然数の足し算をできるのか』という題で講演を行なった．本論文の目的は，この講演の実践例を，配布物の要点と板書のしかたまで含めて紹介することにある．高等学校数学における「数学と人間活動」,「数学と社会生活」，および「数学的な表現の工夫」の話題の参考になれば幸いである．

ノートパソコンなどの計算機は内部にマザーボードと呼ばれる基盤をもち，その基盤上に据えられた中央情報処理装置（CPU）が処理の中枢を担う．現代の実際の CPU は複雑であるが，ここでは CPU を単純化・理想化した数学的な概念としてレジスタマ

シンというものを考える．現実の CPU は 2 進法を用いて効率的に計算を行うが，話を簡単にするため効率は度外視し，2 進法には触れない．

レジスタマシンは正の整数を入れる箱であるプログラムカウンタ（略称 PC）を一つもつ．あとで述べるように，レジスタマシンに仕事をさせるため，人間はマシンにプログラムを与える．プログラムには行番号がついている．PC に格納されている行番号は，これから実行する行を示す．PC 以外に，負でない整数を入れる箱がたくさんあり，これらは汎用レジスタ，あるいは単にレジスタとよばれる．レジスタには番号がついており，第 1 レジスタは R1，第 2 レジスタは R2 とよばれる．第 3 レジスタ以降も同様である．理論上，レジスタは無限個あることにするが，実際にプログラムが一つ固定されたとき，そのプログラムの計算で用いられるレジスタは，そのプログラムごとに決まった特定の有限個である．

現実世界ではアプリ（アプリケーションソフト）の設計図がプログラムであり，プログラムを書くための仕組みがプログラミング言語である．C 言語や Java 言語，C++, Python（パイソン）などがよく知られている．現実のプログラミング言語はいずれも，多くの出来合いの機能（制御構造やライブラリ関数）を持つ．ここでは話を単純化するため，レジスタマシンのプログラムで使える出来合いの機能は必要最小限にしておく．

2　講義序盤で紹介する定義

講演は質疑応答込みで 5 0 分であった．基本的な定義は A4 用紙 1 枚に書いて印刷し配布した．本節では配布物に記した定義を紹介する．レジスタマシンには以下 4 種の基本命令がある．第 1 レジスタ R1 に格納されている非負整数を小文字の r_1 で表す．第 2 レジスタ以降についても同様とする．

レジスタマシンの基本命令

$r_n = 0$　　/* R_n の内容を 0 にする． */

$r_n = r_n + 1$　　/* R_n の内容をインクリメントする（1 増やす）． */

$r_n = r_m$　　/* R_n に R_m の内容を上書きコピーする． */

`if(` r_n == r_m `){ goto k; }`

/* もし R_n と R_m の内容が等しいなら k 行目にジャンプする． */

現実に，たとえば C 言語でプログラムを書くとき，ジャンプ命令は極力使わないのがよいとされる．プログラムが期待通りの動作をしないとき，プログラムの作成者は処理がどこでおかしくなっているかを分析する．プログラム内でジャンプ命令が使われていると，こうした分析がやりにくくなるのである．今述べたのは人間が書いたプ

ログラム，すなわちソースコードの中でジャンプ命令を使わない方がよいという話である．ところでC言語で書いたソースコードは，そのままでは動かない．ソースコードが「アプリの設計図」と呼ばれるのはこのためである．ソースコードをコンパイラ[1]で処理すると機械が実行できる形，いわゆる実行ファイルに変換され，動くようになる．実行ファイルは人間にとっての読みやすさを度外視して，マシンにとっての読みやすさだけを徹底的に追求したものになっている．実行ファイルの中はジャンプ命令にみちあふれている．レジスタマシンのプログラムはソースコードよりも実行ファイルに近い．

基本命令は4種類あると述べたが，4個だとは言わなかった．たとえば第一のパターンとして $r_3 = 0$ や $r_{17} = 0$ は許されるが，$r_n = 0$ は許されない．添字は具体的な正の整数でなければならない．他の3パターンも同様である．初心者相手にこの種の話をするとき，私は必ず「登場人物は二人いると思いましょう」と言うことにしている．「一人はロボット君．つまりマシンの気持ちを代弁する役割です．もう一人はロボット君の上司である人間さんです．人間さんのプロフィールとしては，大学の数学科を卒業した会社員を想像してください．人間さんは初等整数論や，初等整数論を展開するのに必要な程度の初歩的な集合論を常識だと思っています．人間さんは，マシンの動作について考察し，数学的な結論を導き出す担当です．ロボット君にとって，この文脈の添字3や添字17はわかりますが，添字 n は意味不明なのです．添字 n は，人間さんが数学の世界で考えていることなのです」

先頭に行番号を付けた基本命令を有限個並べたものをレジスタマシンのプログラムという．例を示す．

例 **2.1** レジスタマシンのプログラムの例．以下ではこれを P_1 とよぶ．

```
1:  if( r2 == r3 ){ goto 5; }
2:   r1 = r1 + 1
3:   r3 = r3 + 1
4:  if( r1 == r1 ){ goto 1; }
```

プログラムを実行するときの決まりごとは以下の通りである．入力として2個の自然数 a, b を与えると決めた場合，a を第1引数への入力，b を第2引数への入力といい，ベクトルのように書いて入力は (a, b) であるともいう．このとき，a をR1に，b をR2に入れる．時間は離散的に流れ，ロボット君は1行目から順にプログラムを実行

[1] コンパイラもアプリの一種

する．レジスタの内容を書き換える指示があればそれに従う．ジャンプ命令があればそれに従って次に実行する行を決める．ジャンプ命令がなければ，次は一つ下の行を実行する．いつも，今から実行すべき命令は PC にメモしておく．存在しない行を実行しろと指示されたら停止する．停止したとき，R1 に入っている値を出力とみなす．

3　講義中盤の板書の概要

講演の見せ場は，具体的な入力 $(5, 2)$ に対してレジスタマシンの計算を実演するくだりであった．本節では講義中盤の板書のあらましを紹介する．まず横長の黒板の左側にプログラム P_1 を書く．その右隣の列に上から順に PC，R1，R2，R3 と書く．その右隣の列に 4 個のマス目を書く（図 1）．

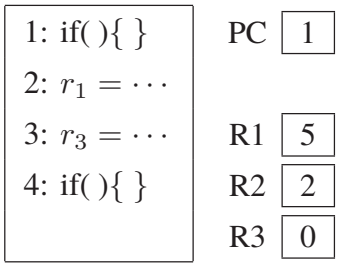

図 1　板書（プログラムおよび初期状態のレジスタ）

PC はプログラムカウンタの略であり，今から実行する行番号を保持する特別な箱である．PC の右のマス目の中には初期状態における PC の内容を書く．1 行目から開始する約束になっているので，このマス目に 1 を入れる．R1，R2，R3 それぞれの隣にあるマス目は初期状態の 1 番，2 番，3 番レジスタである．いま引数は 5 と 2 の二つである場合を考える．第 1 引数「5」を R1 に，そして第 2 引数「2」を R2 に入れる．第 3 レジスタ以降は 0 で埋めておく．今回のプログラム P_1 の中に書かれたレジスタは R1，R2，R3 だけなので，計算の途中で第 4 レジスタ以降への読み書きは生じない．

ここで横長のマグネットを取り出し，今から実行する行すなわち第 1 行の下に取り付ける（図 2）．取り付けるときに「バシッ」と音がするが，聴衆を退屈させない効果があり，ちょうどよい[2)]．図 2 ではマグネットを下線で表している．

[2)] 音に過敏な生徒が教室の最前列にいるとわかっているときはその限りではない．

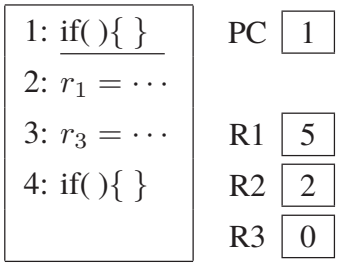

図 2　板書（今から実行する行を横長マグネットで明示した場面）

さっそく教師が「ロボット君」を演じて計算を開始する．第 1 行を確認する．$r_2 = 2 \neq 0 = r_3$ だから goto によるジャンプを行わない．ジャンプしないので PC の値を一つ増やす．すなわち，次の行に進む．マグネットを「バシッ」と動かす．そして右の列に新たなマス目を 4 つ書き足す．一番上の PC には 2 を入れる．他は，直前の作業で変更指示がなかったのでそのままである（図 3）．

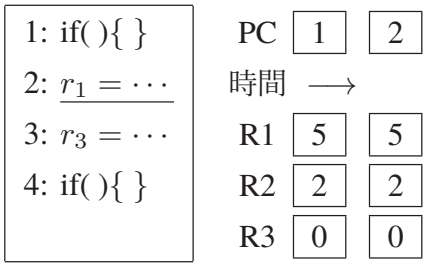

図 3　板書（第 1 ステップ終了後）

r_1 へのインクリメント（値を一つ増やすこと）が指示されているのでそれに忠実に従って第 1 レジスタの内容 5 を 6 に更新する．そして PC の値を一つ増やし，マグネットを一つ下の行に移動する（図 4）．

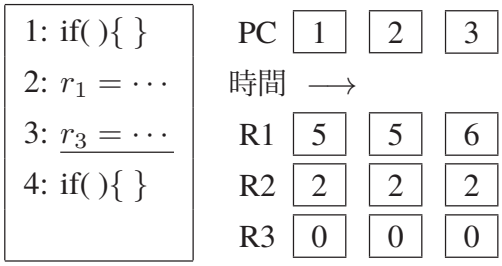

図 4　板書（第 2 ステップ終了後）

人間なら「こんな単純作業をして何になるのだろう」と疑問や不満をもつかもしれないが，マシンにそのような感情はない．第 3 行をよく見てその指示「r_3 のインクリメント」に忠実に従うのみである．第 3 レジスタの内容 0 を 1 に更新し，PC の値を一つ増やし，マグネットを 1 行下へ移動する（図 5）．

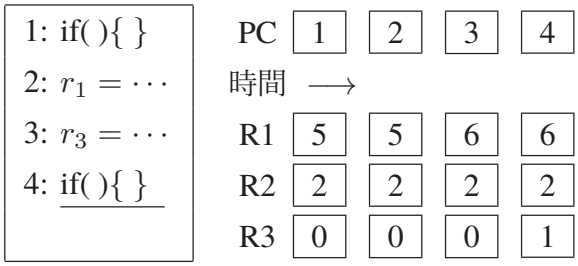

図 5　板書（第 3 ステップ終了後）

第 4 行の指示は「$r_1 = r_1$ ならば第 1 行へジャンプせよ」，すなわち第 1 行への強制ジャンプである．そこで PC の値を 1 に戻し，マグネットを第 1 行に移動する（図 6）．

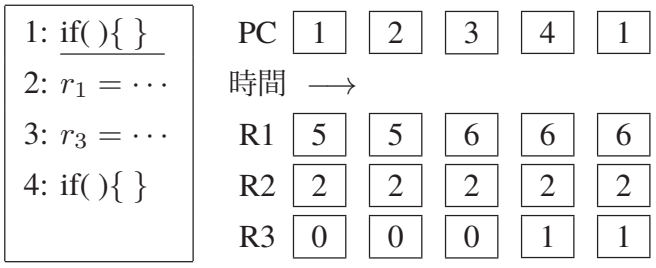

図 6　板書（第 4 ステップ終了後）

再び第 1 行をよく見て指示に忠実に従う．このような場面では，教員がややオーバーアクションで黒板左端のプログラムをよく見ることが大事である．それによって，計算機の中で起きていることへの理解を促すのである．いま $r_2 = 2 \neq 1 = r_3$ だから goto によるジャンプは行わない．以下しばらく，先ほどと似た動作を繰り返す（図 7–10）．

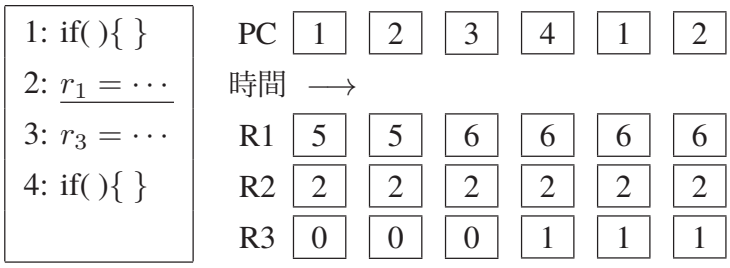

図 7　板書（第 5 ステップ終了後）

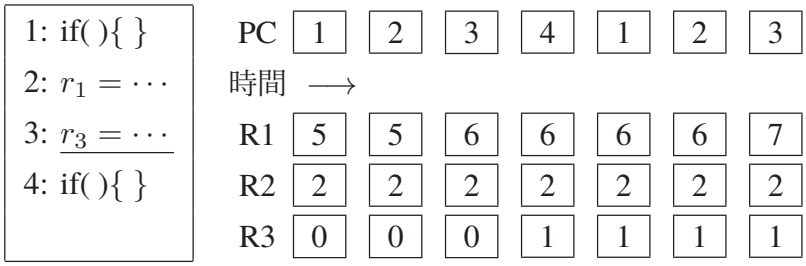

図 8　板書（第 6 ステップ終了後）

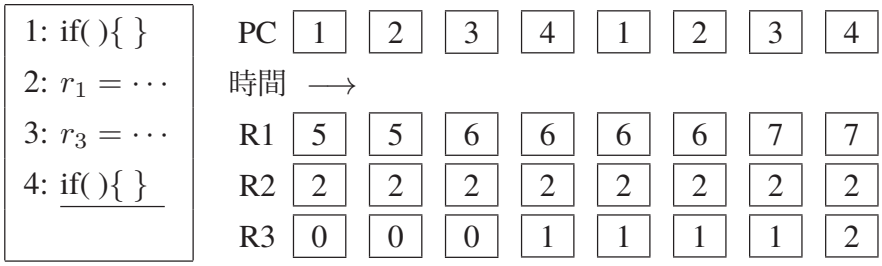

図 9　板書（第 7 ステップ終了後）

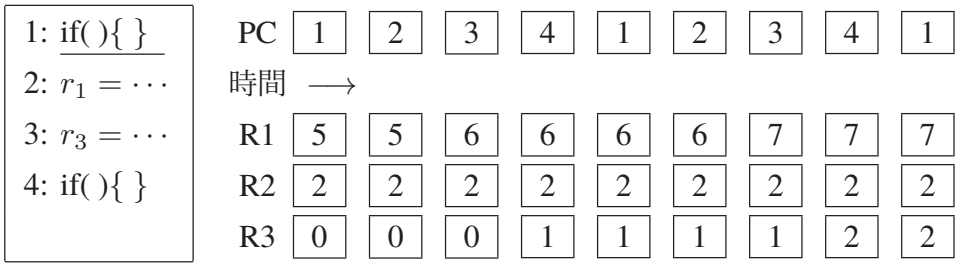

図 10　板書（第 8 ステップ終了後）

今度は $r_2 = 2 = r_3$ となっている．そこでプログラム P_1 の 1 行目にある指示「goto 5」に従って第 5 行目へのジャンプを試みる．ところが第 5 行はない．人間なら疑問をもち，優秀そうな同僚に「どうしたらいいと思う？」と尋ねるかもしれない．しかしマシンはそのように機転を利かすことはない．存在しない行へのジャンプを指示されたらそこで停止する（図 11）．このとき R1 の内容は 7 だから，7 を出力して停止したことになる．

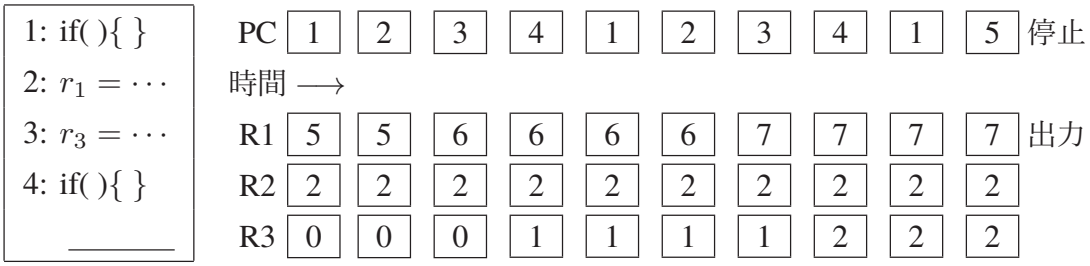

図 11　板書（第 9 ステップ終了後）

4　講義終盤の概要

ここで教師は「ロボット君」とは別の登場人物「人間さん」を演じ，「いったいこの計算には何の意味があるのだろう」と問いかける．そして計算終了後の図を少し書き換え，上記で $5+2$ が出力されたことを観察する（図 12）．

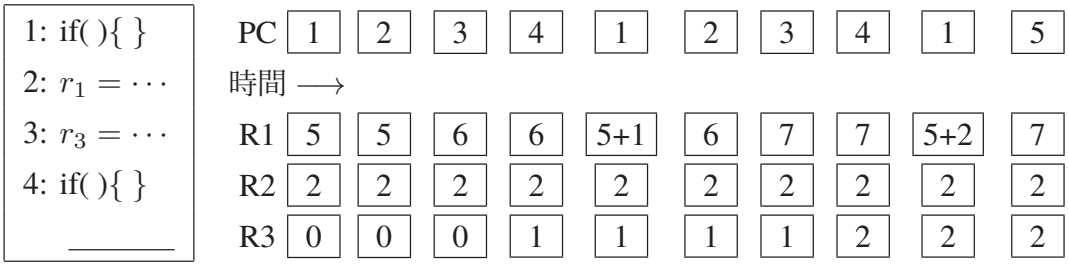

図 12　板書（計算終了後の考察）

次に「では入力が $(5,2)$ ではなく，(a,b) のときはどうなるのだろう」と問いかける．必要に応じて聴衆と言葉を交わしながら，図を書き換えて考察する．

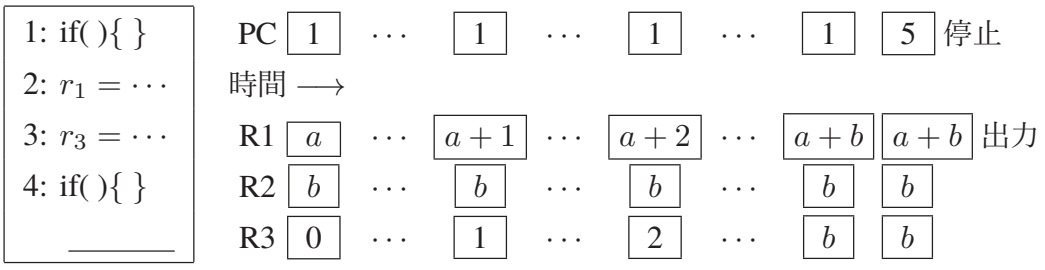

図 13　板書（一般の場合の考察）

「おや，不思議ですね．どんな入力 (a,b) に対しても，必ずこのプログラムは有限回で停止するし，しかも停止したときには必ず $a+b$ を出力しますよ．いま大雑把に図を書いて考察しましたが，このことをきちんと証明しようとしたらどうしますか」教

師は「人間さん」を演じながらこのように言って，聴衆に問いかける．聴衆から「数学的帰納法」という言葉が出ればありがたい．

人間さん「ロボット君，すごいじゃないか．君は足し算ができるんだね」

ロボット君「タシザン？ それは何でありますか．自分は，忠実に命令に従っただけであります」

講演のしめくくりに，以下のように話す．「こうしてみると，マシンにとって知性とは何か，というのは面白い奥深い問いのようですね．最初に入力を受け取った後，ロボット君がやったことと言えば，プログラムを読みに行く，そしてそこで指示されたことと直前の時点で箱に入っていた数を使って箱の中身を更新する，ひたすらそれを繰り返すだけでした．たったそれだけで，計算を生み出し，足し算を実行できたのです．ところで皆さんは高校数学で似たようなアイデアを学んだことがあるはずです．初項を受け取った後，漸化式の指示と，直前に作った第 n 項の値を用いて第 $n+1$ 項を計算する，ひたすらそれを繰り返すだけで数列を生み出せたのです．数学は直接社会の役立つだけでなく，数学の考え方が形を変えて社会の役に立つことも多いのです．計算機の頭脳のど真ん中で，漸化式と似たアイデアが活用されているのです」

漸化式の話は最後にさりげなく出てきただけであるが，聴衆の人間心理としては，こういう言い方をされた方が漸化式の意義が心に残ると思われる．

5 結び

計算機は数学学習の役に立つ．このことは平成３０年告示高等学校学習指導要領でも強調されている．同指導要領の教科「数学」の部分 [文科省 2019, pp.91–102] には「コンピュータなどの情報機器を用いて」と「コンピュータなどの情報機器を用いる」という表現が合計 7 箇所も出現し，これらとは別に「コンピュータや情報通信ネットワークなどを適切に活用」という表現も出現する．本講演に込めた暗黙のメッセージは「逆に，数学は計算機システムやアルゴリズムのデザインに役立つ」ということであり，さらに踏み込んで言えば「数学とコンピュータはお互いに助け合う双方向的な関係にある」ということである．

筆者は計算可能性理論のごく初歩を学部 3 年生，4 年生向けに講義しており，第 1 回の授業の中に，上記のレジスタマシンの話を盛り込んでいる．当該授業のレジュメはネットで公開しており，[鈴木 2012] はその一部である．本稿執筆にあたり [鈴木 2012] を参考にした．さらに，[鈴木 2012] のレジスタマシンの部分を執筆する上で Cooper [Co2004] を参考にした．これは計算可能性理論の専門書である．筆者は大学の学部 4 年生や大学院生のゼミのテキストとしてしばしばこれを用いた．

【参考文献】

[Co2004] Cooper, S. B., *Computability theory*. Chapman and Hall/CRC (2004).

[鈴木 2012] 鈴木 登志雄, 「計算可能性理論（序）」ウェブ上のプリント (2012). http://www.comp.tmu.ac.jp/mathsuzuki/classroom.html

[文科省 2019] 文部科学省, 「高等学校 学習指導要領（平成３０年告示）」MEXT 1 — 1812 東山書房 (2019).